In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown.

1. (Currently Amended) A method comprising:

inserting a single instruction at a start of a block of code to run on a first processor ~~to determine if resources of a processor are available for the block of code, wherein the block of code includes multiple instructions~~;

emulating the block of code on a second processor;

using the single instruction to monitor the resources of the second processor used during emulation to determine whether resource requirements of the first processor have been exceeded; and

if the ~~resources~~ resource requirements of the first processor have been exceeded ~~are available~~, modifying ~~the available~~ allocation of the resources of the second processor according to the resource requirements of the first processor ~~multiple instructions in the block of code~~.

2. (Previously Presented) The method of claim 1, further comprising:

determining a set of available resources that will be available after said block of code has executed.

3. (Canceled)

4. (Currently Amended) The of claim 1 wherein the availability of the processor resources ~~are~~ is determined at compile time.

5. (Currently Amended) The method of claim 1 wherein the availability of the processor resources ~~are~~ is determined dynamically.

6. (Previously Presented) The method of claim 1 further comprising: signaling an error message if the resources of the processor needed for the block of code are not available; and

in response to the error message, branching to a fault handler routine.

7. (Previously Presented) The method of claim 6 wherein signaling of said fault handler routine simulates a processor exception.

8. (Previously Presented) The method of claim 1 wherein the resources are represented by a bit vector.

9. (Previously Presented) The method of claim 8 wherein said bit vector is generated dynamically.

10. (Currently Amended) A computer-readable medium having stored thereon a set of instructions to monitor processor resources, said set of instruction, which when executed by a processor, cause said processor to perform a method comprising:

inserting a single instruction at a start of a block of code to run on a first processor ~~to determine if resources of a processor are available for the block of code, wherein the block of code includes multiple instructions~~;

emulating the block of code on a second processor;

using the single instruction to monitor the resources of the second processor used during emulation to determine whether resource requirements of the first processor have been exceeded; and

if the ~~resources~~ resource requirements of the first processor have been exceeded ~~are available~~, modifying ~~the available~~ allocation of the resources of the second processor according to the resource requirements of the first processor ~~multiple instructions in the bock of code~~.

11. (Currently Amended) The computer-readable medium of claim 10, wherein said set of instructions further includes additional instructions, which when executed by said processor, cause said processor to further perform said method ~~further~~ comprising:

determining a set of available resources that will be available after said block of code has executed.

12. (Canceled)

13. (Currently Amended) The computer-readable medium of claim 10 wherein the availability of the processor resources ~~are~~ is determined at compile time.

14. (Currently Amended) The computer-readable medium of claim 10 wherein the availability of the processor resources ~~are~~ is determined dynamically.

15. (Previously Presented) The computer-readable medium of claim 10 wherein additional instructions, which when executed by the processor, cause the processor to perform the method further comprising:

signaling an error message if the resources of the processor needed for the block of code are not available; and

in response to the error message, branching to a fault handler routine.

16. (Previously Presented) The computer-readable medium of claim 15 wherein signaling of said fault handler routine simulates a processor exception.

17. (Previously Presented) The computer-readable medium of claim 10 wherein the resources are represented by a bit vector.

18. (Previously Presented) The computer-readable medium of claim 17 wherein said bit vector is generated dynamically.

19. (Currently Amended) A computer-readable medium, having stored thereon a first set of instructions, the first set of instructions, which when executed by a processor, ~~generate~~ generates a second set of instructions through a binary translation process, the second set of

instructions when executed by the processor, ~~cause~~ causes said processor to perform a method comprising:

inserting a single instruction at a start of a block of code to run on a first processor ~~to determine if resources of a processor are available for the block of code, wherein the block of code includes multiple instructions~~;

emulating the block of code on a second processor;

using the single instruction to monitor the resources of the second processor used during emulation to determine whether resource requirements of the first processor have been exceeded; and

if the ~~resources~~ resource requirements of the first processor have been exceeded ~~are available~~, modifying ~~the available~~ allocation of the resources of the second processor according to the resource requirements of the first processor ~~multiple instructions in the bock of code~~.

20. (Currently Amended) The computer-readable medium of claim 19, wherein said set of instructions further includes additional instructions, which when executed by said processor, cause said processor to further perform said method ~~further~~ comprising:

determining a set of available resources that will be available after said block of code has executed.

21. (Canceled)

22. (Currently Amended) The computer-readable medium of claim 19 wherein the availability of the processor resources ~~are~~ is determined dynamically.

23. (Previously Presented) The computer-readable medium of claim 19 wherein additional instructions, which when executed by the processor, cause the processor to perform the method further comprising:

signaling an error message if the resources of the processor needed for the block of code are not available; and

in response to the error message, branching to a fault handler routine.

24. (Previously Presented) The computer-readable medium of claim 23 wherein signaling of said fault handler routine simulates a processor exception.

25. (Previously Presented) The computer-readable medium of claim 19 wherein needed processor resources are represented by a bit vector.